MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo 826 January, 1985

Mobile Robot Localization Using Sonar

Michael Drumheller

Abstract. This paper describes a method by which range data from a sonar or other type of rangefinder can be used to determine the 2-dimensional position and orientation of a mobile robot inside a room. The plan of the room is modeled as a list of segments indicating the positions of walls. The method works by extracting straight segments from the range data and examining all hypotheses about pairings between the segments and walls in the model of the room. Inconsistent pairings are discarded efficiently by using local constraints based on distances between walls, angles between walls, and ranges between walls along their normal vectors. These constraints are used to obtain a small set of possible positions, which is further pruned using a test for physical consistency. The approach is extremely tolerant of noise and clutter. Transient objects such as furniture and people need not be included in the room model, and very noisy, low-resolution sensors can be used. The algorithm's performance is demonstrated using a Polaroid Ultrasonic Rangefinder, which is a low-resolution, high-noise sensor.

Acknowledgments. This report describes research done in part at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology and in part at Thinking Machines Corporation, Cambridge, MA. Support for the Laboratory's Artificial Intelligence research is provided in part by a grant from the System Development Foundation, and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014-80-C-0505 and N00014-82-K-0334.

1 Problem Definition

The specific problem considered in this paper is that of enabling a mobile robot to determine its position and orientation (its *configuration*) inside a building in a way independent of assumptions about its previous movements. This ability will be called *absolute localization*, or simply *localization*. Localization is the direct measurement of vehicle position. It is to be contrasted with *dead-reckoning*, or *trajectory integration*, which is the process of measuring a vehicle's velocity relative to nearby stationary objects (the ground, for example), and deducing the vehicle's position from its velocity history.

Since the errors encountered in dead-reckoning are cumulative, a robot that navigates by dead-reckoning alone will eventually lose track of its position. Ultimately, this can be prevented only by periodically re-establishing the absolute position of the robot. Therefore a means of localization is necessary for safe, reliable robot navigation. Mobile robots that will someday be operating in factories, offices, homes, hospitals, etc., will need a reliable means of localization.

Some solutions to this problem have been proposed that require modifications to the environment, such as triangulation from infra-red beacons [Giralt, Sobek, & Chatila 79]. It would be desirable to solve the problem without modifying the environment. Furthermore, most of the mobile robot navigation schemes developed so far, such as in [Moravec 81], are essentially dead-reckoning methods, which lack any provision for periodically localizing the robot. Such schemes could benefit from the addition of a means of localization.

In the localization approach described in this paper, the robot's environment is a room or area inside a building. The environment could include the whole building. The robot's user must provide a model of the room consisting of a list of segments indicating the locations of walls. Such a model is easily constructed from an architect's drawing or with a tape measure.

The rangefinding device used in this paper is a Polaroid Ultrasonic Rangefinder, but any rangefinder may be used (see, for example, [Jarvis 83] and [Thompson 79]). We will henceforth call ultrasonic rangefinding *sonar* for short.

The closed contour obtained by a 360-degree sweep with a sonar beacon will be called a *sonar contour*. The lines drawn from the robot position to the individual data points in the sonar contour represent individual range readings, and are called *sonar rays*.

Figure 1 shows a typical room outline, a typical sonar contour obtained from inside the room, and the corresponding localization as determined by the algorithm.

2 Approach

Some recent papers ([Grimson & Lozano-Pérez 83] and [Gaston & Lozano-Pérez 84]), have introduced a new approach to object recognition and localization based on exploiting simple geometric constraints between sensed data and a model. Sections

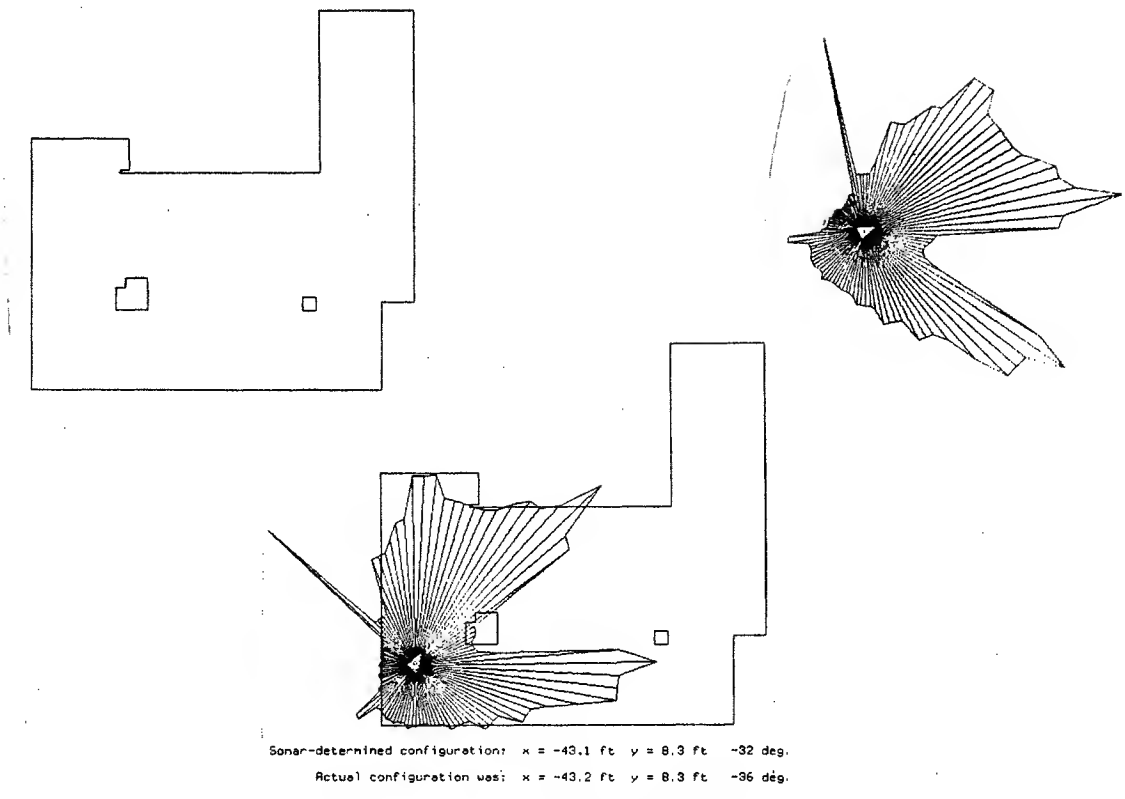


Figure 1: *A typical room, a sonar contour obtained from inside the room, and the localization produced by the algorithm. Note the printed numerical results*

2.4, 2.5, and 2.7 are based largely on the object recognition method described in [Grimson & Lozano-Pérez 83, 84] and [Gaston & Lozano-Pérez 84]. The main difference between these sections and these papers is the use of *sonar segments*, which are straight segments extracted from a sonar contour by a simple line-fitting algorithm, instead of position/normal-vector pairs, as the primary inputs to the algorithm.

This paper introduces a new idea, called the *sonar barrier test*, in section 2.6. The sonar barrier test checks for physical consistency by determining whether the shape of a sonar contour for a proposed localization is consistent with the simple fact that sonar beams do not penetrate solid objects. If an inconsistency is found, the proposed localization is discarded. The sonar barrier test makes possible overall algorithm performance that is superior to what was obtainable using only the methods described in [Grimson & Lozano-Pérez 83, 84] and [Gaston & Lozano-Pérez 84].

[Miller 84] also describes an approach to robot localization using sonar, following the methods of [Grimson & Lozano-Pérez 83, 84] and [Gaston & Lozano-Pérez 84]. The method described in [Miller 84] uses single sonar rays instead of larger data features as the primary inputs to the algorithm, and it uses a different set of local geometric constraints. Nothing analogous to the sonar barrier test is presented, and only one experimental result is shown.

Our approach is to consider the localization process to be a 2-dimensional matching (including rotation) between the sonar contour and the room outline. We wish

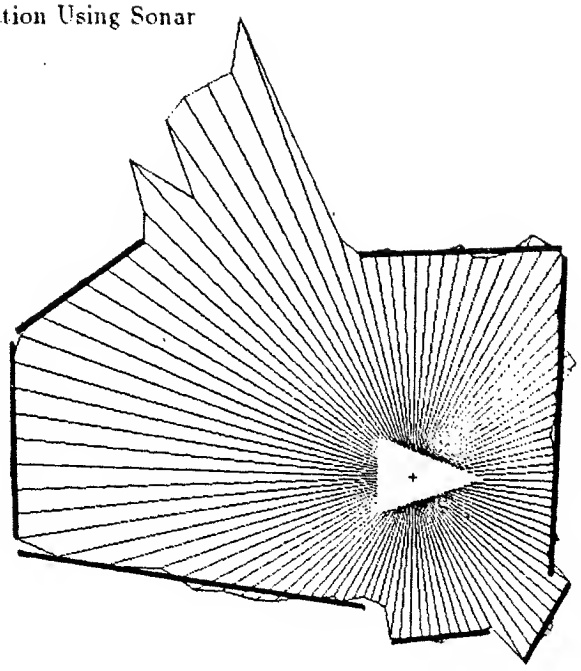


Figure 2: *Some straight segments extracted from a typical sonar contour.*

to determine the geometrical relationship between the robot and the room. The configuration of the robot relative to the sonar contour is always known, so if we can determine a possible configuration for the sonar contour relative to the room outline, then we will have found a possible configuration for the robot inside the room.

The goal of the localization process, therefore, is to find possible matches of the sonar contour to the room outline. We proceed in four steps:

1. **Extract straight line segments from the sonar data**: Straight segments extracted from a sonar trace are called *sonar segments*. An example of some sonar segments extracted from a sonar contour is shown in Figure 2. The matching process is initially driven entirely from the sonar segments, which are usually the sonar images of walls.

2. **Generate feasible interpretations**: A set of feasible *interpretations* of the sonar segments is constructed. An interpretation is a set of ordered pairs of sonar segments and walls, where $[(seg_i\ wall_{w_i})\ (seg_j\ wall_{w_j})\ldots]$, means "it is feasible that seg_i could be the sonar image of $wall_{w_i}$, seg_j could be the sonar image of $wall_{w_j}$, etc.." Interpretations that are inconsistent with local constraints (derived from the model) on the sonar segments are discarded.

3. **Global Model Test**: The feasible interpretations are tested for consistency with the equations of the walls in the model. An interpretation is admissible if it is possible to find a rotation and translation of the sonar contour that would superpose each sonar segment over the wall with which it is paired, while keeping the sensor inside the room and the sonar segment endpoints within the limits of the wall endpoints.

4. **Sonar Barrier Test**: An interpretation may represent a geometrically feasible configuration for the sonar segments alone, but an impossible configuration for the entire sonar contour. In particular, each interpretation that survives the global model test must also pass the *sonar barrier test*, namely: an admissible robot configuration must not imply that any sonar ray penetrates a known solid object.

The second step is the key to the process. The number of possible interpretations, given s sonar segments from the sonar contour and n walls in the model, is n^s. Since the global model test and the sonar barrier test are computationally expensive processes, it would be impractical to perform each of them on all possible interpretations. [Grimson 84] shows that the number of feasible interpretations can be reduced to manageable numbers through the use of local geometric constraints.

2.1 The Rangefinder and Rangefinding Error

The Polaroid Ultrasonic Rangefinder was chosen for this research because of its simplicity, availability, and low cost. It consists of an ultrasonic transmitter, a microphone, and a timing mechanism. The transmitter and microphone functions are performed by a single physical transducer. Range information is obtained by broadcasting a pulse of ultrasound and measuring the elapsed time until an echo is received. The Polaroid Ultrasonic Rangefinder is described in detail in [Polaroid]. Other suitable rangefinding devices are described in [Massa] and [Jarvis 83].

There are several sources of error that can corrupt sonar range data. These are discussed in the following paragraphs.

2.1.1 Errors Due to the Shape of the Sonar Beam

The beam pattern of the Polaroid device is similar to the complicated multi-lobed pattern produced by any circular-disk acoustic emitter. For simplicity, however, we model the beam as cone-shaped. (See Figure 3.)

When the Polaroid device is aimed perpendicular to a flat surface it reports the true range to the surface to within about one-half of an inch. However, the range error can be much larger when the beam strikes a surface with a large angle of incidence. The reason is that the *edge* of the wavefront is reflected back to the sensor instead of the *centerline* (see Figure 4). This effect, called *radial error*, often results in errors greater than one foot.

Because of the large beam width (the beam has a half-angle β of about 15 degrees), the rangefinder tends to produce a blurred image of its surroundings. This effect, called *angular error*, is similar to convolving a range contour obtained by a perfect rangefinder with a pulse whose width is proportional to the range being measured.

For simplicity, we will henceforth lump radial and angular errors together, modelling their combination as unpredictable but bounded by a constant E; thus we assume that *the location of the endpoint of any particular sonar ray may be in error*

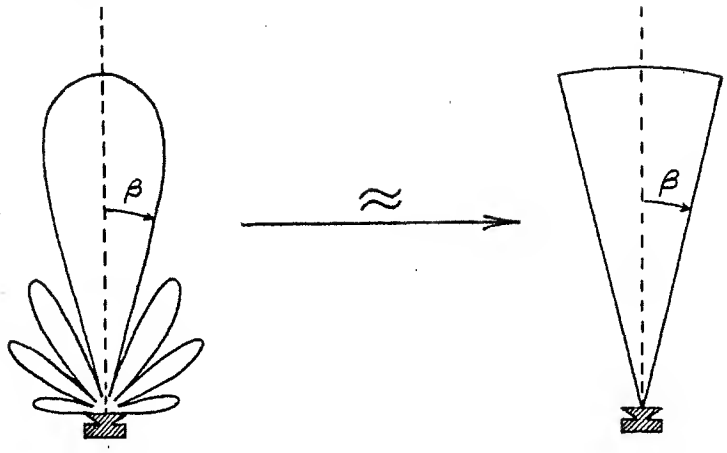


Figure 3: *Cone-shaped approximation of actual multi-lobed beam pattern.*

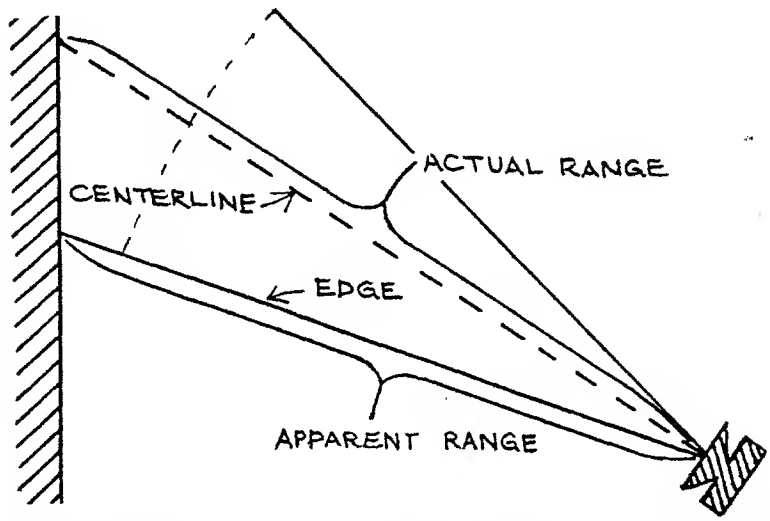


Figure 4: *Ranging error can be caused by edge-of-beam reflection.*

by as much as E. The actual value of E, which is determined empirically, will be discussed in the Results section.

2.1.2 False Reflections

Based upon the preceding discussions, one might think it possible to extract, with reasonable accuracy, the room outline from the sonar contour by means of a deconvolution process. However, the nature of ultrasound reflections makes this virtually impossible.

The pulse emitted from the Polaroid device has a frequency of about 55 kHz and a wavelength of about a quarter of an inch. Therefore, unless the sensed surface has irregularities whose size is of the same order, the sonar beam will not be scattered. It may bounce off into oblivion after striking the surface at a large angle of incidence, instead of reflecting a strong echo back to the sonar receiver. This effect, called *false reflections*, occurs whenever the *incidence angle* of the beam, called ψ, is greater than

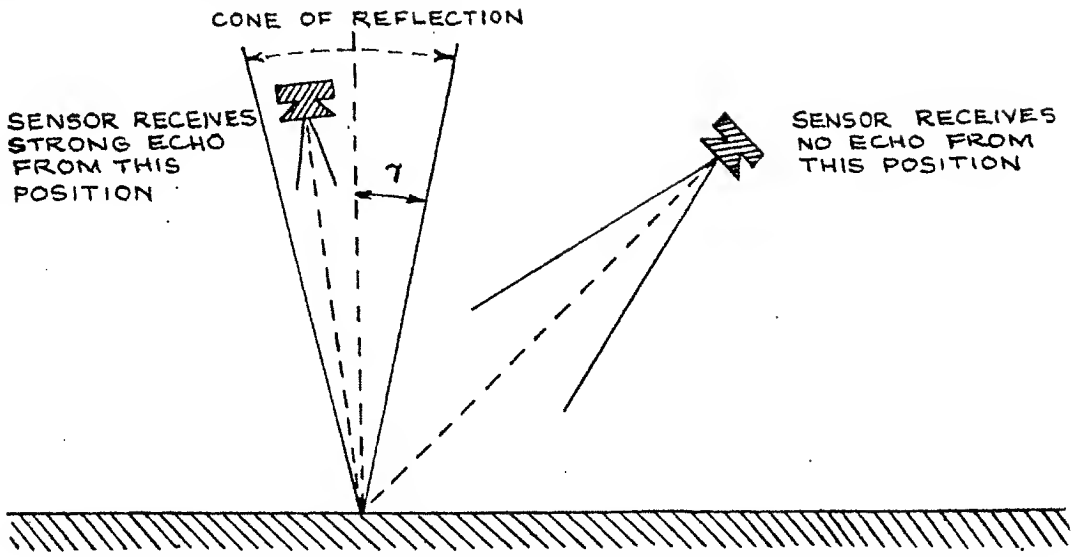


Figure 5: *The Cone of Reflection, with half-angle* γ.

a critical angle called γ, which defines the *cone of reflection* (CR) for the surface (see Figure 5). A sonar beam striking a wall from inside the CR will reflect back to the sensor with sufficient strength to produce an accurate range reading. A beam striking the wall from outside the CR will be reflected away from the sensor, producing an unrealistically long sonar ray. The sonar beam thereby apparently penetrates the wall. An example of this effect may be seen as the sharp "horn" jutting out of the sonar contour in Figure 1.

Every surface material has its own CR half-angle, which may range from seven or eight degrees (for glass) to nearly ninety degrees for rough surfaces.

2.2 Clutter

Another source of error that the localization method must overcome is *clutter*. Clutter is *any object that is not included in the room model*. Clutter often distorts sonar contours so much that they bear almost no overall resemblance to the room outline. We will show that the approach described here is very effective at ignoring clutter.

2.3 Extracting Straight Line Segments

A sonar contour consists of 100 range readings, taken from a single position in the room, at 3.6-degree angular intervals. The sensor was mounted at an altitude of 5.5 feet on a stepper-motor driven "head," which could position the sensor under computer control.

The straight-line finder used for extracting straight segments from a contour is the "iterative endpoint fit," described in [Ballard & Brown 82]. This algorithm extracts the *contiguous non-overlapping subcontours* from a sonar contour that best approximate straight segments. (A contiguous subcontour is a group of sonar ray

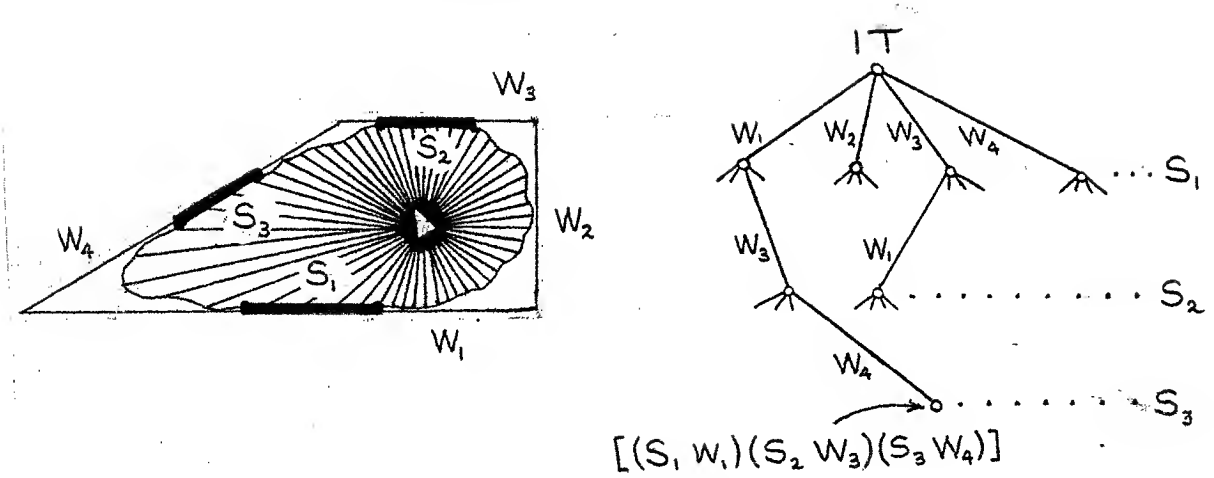


Figure 6: *An example of an interpretation tree for a 4-wall room model and a 3-segment sonar contour.*

endpoints that occur consecutively in the sonar contour.) The algorithm has two operating parameters, N and δ; it finds the set of longest non-overlapping contiguous subcontours containing at least N points, and having no point farther than δ from the line passing through the endpoints of the subcontour.

2.4 Generating Feasible Interpretations

After obtaining a sonar contour and extracting the sonar segments, we have up to s sonar segments S_i, and we seek a geometrically consistent pairing of these sonar segments with some number k of the n walls that comprise the model of the building. For now we will assume that all the sonar segments are sonar images of walls. (This is not necessarily the case, since a piece of furniture or other clutter, or even sensor noise, may give rise to a sonar segment that cannot be interpreted as the sonar image of any wall. We will address this issue later in this paper.) The range of possible interpretations can now be cast in the form of an *interpretation tree* (IT), an example of which is shown in Figure 6. The root node of the IT has n descendants, each representing an interpretation in which S_1 is the sonar image of a different wall in the room model. There are a total of s levels in the tree. A node at level i indicates one set of possible pairings of sonar segments S_1 through S_i with the walls of the room.

2.5 Pruning the IT Using Local Constraints

Only a small number of the interpretations in the IT are geometrically consistent with the sonar contour. We can exploit the following local constraints to prune the inconsistent interpretations:

1. **Distance Constraint**—The range of possible distances between points on a pair of sonar segments, taking sensor error into account, must overlap the

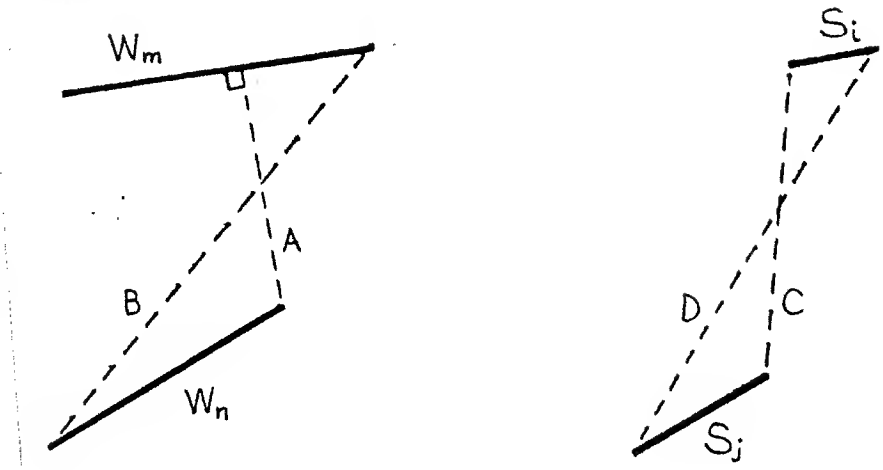


Figure 7: *The range of distances between wall pairs and sonar segment pairs.*

range of distances between the pair of walls that they are assigned to in an interpretation.

Consider the pair of walls W_m and W_n and the pair of sonar segments S_i and S_j in Figure 7. The range of distances between W_m and W_n is $[A, B]$, and the range of distances between S_i and S_j is $[C, D]$, where A, B, C, and D are defined by Figure 7 for this situation. (A, B, C, and D can be calculated easily for any pairs of sonar segments and walls.) However, the range $[C, D]$ does not take sensor error into account. Using our simple bounding-constant model of rangefinder error, the range of possible distances between S_i and S_j becomes $[(C \pm 2E), (D \pm 2E)]$.

Therefore, if an interpretation assigns S_i to W_m and S_j to W_n, then it must be true that $(D - 2E) \leq B$ and $(C + 2E) \geq A$ for the interpretation to be feasible.

2. **Angle Constraint**—The range of possible angles between a pair of sonar segments, taking sensor error into account, must include the known angle between the pair of walls that they are assigned to in an interpretation.

   Figure 8 shows that since any sonar measurement can be in error by as much as E, then any sonar segment S_i may be regarded as (possibly) the image of some straight object having a minimum length d_i and occupying any position inside the dashed line surrounding S_i. The orientation of S_i can be seen to be in error by as much as ϕ_i. Thus the range of possible angles between two sonar segments S_i and S_j is $[(\theta - (\phi_i + \phi_j)), (\theta + (\phi_i + \phi_j))]$. If the known angle between W_m and W_n is α_{mn}, then an interpretation assigning S_i to W_m and S_j to W_n is feasible only if $(\theta - (\phi_i + \phi_j)) \leq \alpha_{mn} \leq (\theta + (\phi_i + \phi_j))$.

   In practice, the above geometrical construction for computing the angular error of a sonar segment is unrealistic for sonar segments whose length is of the same order as E. Therefore, it is useful to place an upper bound E_ϕ on the angular error for a sonar segment. The actual value of E_ϕ, which appears in the Results section, is determined empirically by estimating its value in several

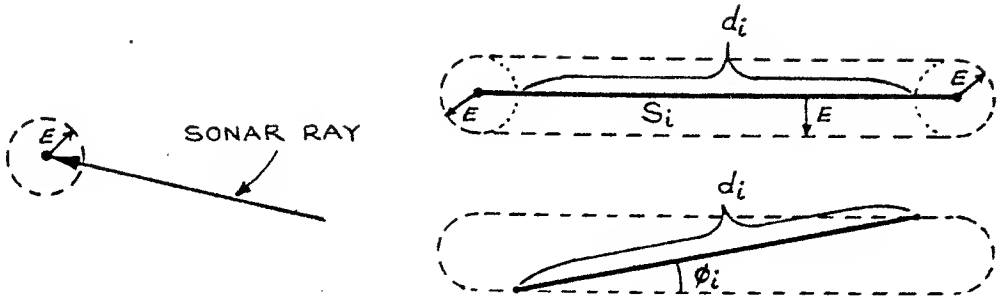


Figure 8: *Calculating the angular error ϕ_i for a sonar segment S_i: S_i is possibly the image of a straight object of minimum length d_i, occupying any position inside the dashed line. Therefore, the orientation of S_i may be in error by as much as ϕ_i, as shown.*

sonar contours.

Note that any wall or sonar segment P_i has an inward-pointing normal vector, $\hat{\mathbf{n}}_{P_i}$, whose positive direction is toward the inside of the room or toward the sensor. It is thus natural to define the angle between two walls or sonar segments as the inner product of their normal vectors. However, this definition provides less geometric constraint than is really available. Consider Figure 9. The inner product of the normal vectors of sonar segments S_a and S_b is the same as that for walls W_x and W_y. It is clear from the figure, however, that it would not make sense to assign S_a to W_x and S_b to W_y. For this reason, we define the *directed angle* between two walls or sonar segments P_i and P_j to be the pair of numbers (A_{ij}, B_{ij}), where A_{ij} is the component of P_i's inward-pointing normal vector $\hat{\mathbf{n}}_{P_i}$ in the direction of $\hat{\mathbf{n}}_{P_j}$, and B_{ij} is the component of $\hat{\mathbf{n}}_{P_i}$ in the direction to the right of $\hat{\mathbf{n}}_{P_j}$. Defined in this way, the directed angle between S_i and S_j is not equal to that between W_x and W_y.

3. **Normal-Direction Constraint**—This constraint is most easily explained using an illustration (see Figure 10). Sonar segments S_g and S_h survive the distance- and angle-constraints for the interpretations $[(S_g\ W_q)(S_h\ W_r)]$ and $[(S_g\ W_q)(S_h\ W_s)]$. It is clear, however, that only the latter interpretation is geometrically consistent. This is because the range of distances between a pair of sonar segments S_i and S_j *in the direction of each sonar segment's inward-pointing normal vector* must be consistent with the similar range for any pair of walls W_m and W_n. In Figure 10 we can see that the range of distances from S_g to S_h along $\hat{\mathbf{n}}_{S_g}$ is consistent with the range of distances from W_q to W_r along $\hat{\mathbf{n}}_{W_q}$, but it is not consistent with the similar range from W_q to W_s.

This "normal direction range" is computed straightforwardly for pairs of walls. The position error E and the angular error E_ϕ must both be included when computing this range for a pair of sonar segments. This may be quickly approximated by considering the maxima and minima of the normal ranges occurring

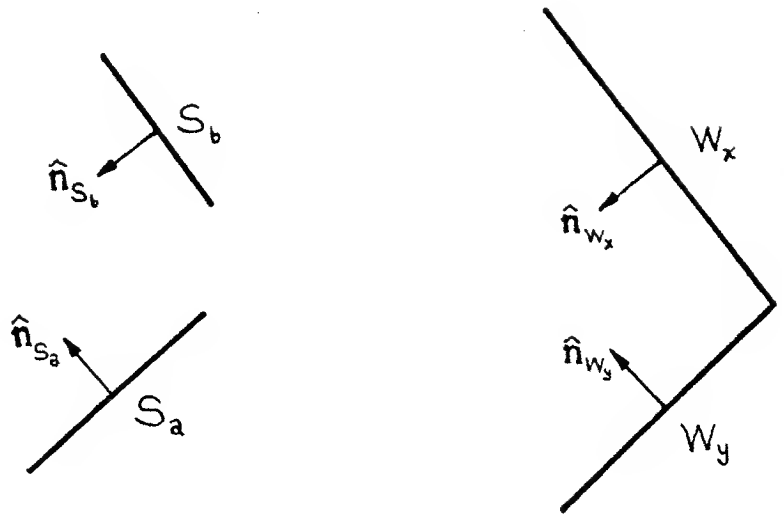


Figure 9: *A situation illustrating the usefulness of the* directed angle *concept.*

in a few extreme cases of possible sonar segment orientations and positions relative to walls.

The local constraints described above are described in greater detail in [Gaston & Lozano-Pérez 83], [Grimson & Lozano-Pérez 83], [Grimson 84], and [Drumheller 84].

Using these constraints, the IT is expanded in the following way: At the first level in the IT, S_1 is allowed to be paired with each wall that is longer than it, since none of the above local constraints (which are all pairwise) applies to solitary sonar segments. Below the $[(S_1\ W_i)]$ node at the second level, S_2 is paired with each wall W_j that is consistent, based on the local constraints, with the pairings $[(S_1\ W_i)(S_2\ W_j)]$. Below the node $[(S_1\ W_i)(S_2\ W_k)]$ at the third level, S_3 is paired with each wall W_l that is consistent with the pairings $[(S_1\ W_i)(S_3\ W_l)]$ and $[(S_2\ W_k)(S_3\ W_l)]$. Note that for a segment-wall pairing to be consistent, it must be consistent with every pairing that leads to it on a path through the IT. Thus, for a three-segment interpretation the local constraints must be applied to three segment-wall pairings, and in general $\binom{k}{2}$ pairings for k sonar segments.

The above constraints will usually prune all but a handful of the non-feasible interpretations from the IT. The pruning often occurs very early in the generation process, eliminating large subtrees from consideration. It is important to note that the constraints will not generally reject all impossible interpretations. Suppose, for example, that the interpretation $[(S_i\ W_m)(S_j\ W_n)]$ easily passes the distance constraint, even though S_i and S_j must eventually be situated within two relatively small intervals on W_m and W_n. This restriction on the positions of S_i and S_j is ignored at further generation steps, since further sonar segments will be required to be distance-consistent with the full lengths of W_m and W_n, instead of the small intervals implied by the previous pairings. In order to guarantee that all the available geometric constraints on the sonar segments are used, we use the global model test,

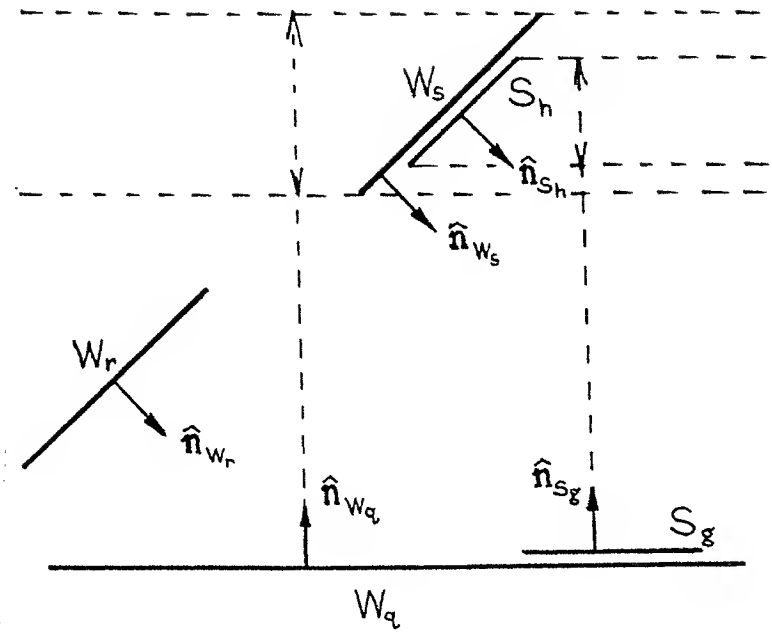


Figure 10: *The geometry of the normal direction constraint.*

which is described later.

Of course, the above constraints are not the only ones that may be used in this kind of application. They were chosen because they are very simple to implement, and because similar ones have been shown to be very effective despite their relative weakness; refer to [Grimson 84] and [Grimson & Lozano-Pérez 83] for results that demonstrate this point.

2.5.1 The Global Model Test

In the global model test we seek, for each surviving interpretation, a translation and rotation of the sonar contour that will superpose each sonar segment onto the wall with which it is paired. In this paper we use a simple averaging scheme as follows:

To find the orientation of the sonar contour, we assume that each segment lies flat against its assigned wall. Thus, each segment-wall pairing implies a particular orientation for the sonar contour with respect to the room. The average of these orientations is taken to be the actual value of the orientation of the sonar contour.

To determine the proper translation, we take each pair of pairings in the interpretation and determine the translation that causes the midpoints of the sonar segments to intersect their assigned walls, given the sonar contour orientation determined above. (A compact algorithm for computing this translation appears in [Grimson & Lozano-Pérez 83].) The average of these translations is taken as the actual value of the translation of the sonar contour. Note that a pair of pairings in which both sonar segments or both walls are parallel to each other does not help to locate the robot uniquely. Therefore, the translation contribution from such pairs is

ignored.

Once we have determined the configuration implied by a particular interpretation, we check to be sure the endpoints of each sonar segment lie within E of the wall with which it is paired. We must also check to be sure that each sonar segment lies within E of the finite bounds of these walls. (Since the walls are described as segments, not infinite lines, it is possible that some sonar segments may be transformed to lie beyond the endpoints of the walls, which would disqualify an interpretation.)

2.5.2 The Inside Test

The global model test includes a simple check to make sure that the interpretation under consideration localizes the robot to lie *inside* the room. This procedure uses the well known fact that a point lies inside a closed contour if and only if ray drawn from the point to infinity intersects the contour an odd number of times.

2.6 The Sonar Barrier Test

There is one final test that each interpretation surviving the global model test must pass: An admissible interpretation must not imply that the sonar beam penetrates any walls from inside their cone of reflection.

Figure 11 illustrates this point. After we have performed the global model test we are left with a set, which may be empty, of possible sonar contour (and thus robot) configurations. For example, in Figure 11, each proposed localization may have survived all of the local constraints and even the global model test.

However, only the lower left configuration in Figure 11 is physically possible, since it is the only one that does not imply that the sonar beam penetrates a solid wall from inside the cone of reflection for that wall. Note that admissible interpretations often imply that the beam penetrates a wall from outside the CR for that wall. This situation is perfectly acceptable, since it does not violate any physical laws.

Therefore, whenever we find an interpretation that passes the global model test, we perform the *sonar barrier test* as follows:

- Attempt to find a sonar ray that
  - intersects any wall,
  - lies within the cone of reflection for the wall,
  - has its endpoint outside the wall by more than a small amount;
- If any such ray exists, then discard the interpretation and the localization, since they imply a physically impossible situation.

Note that the amount by which a sonar ray must lie outside the wall is not necessarily E. We can afford to use a smaller error bound, called E_n (for "normal

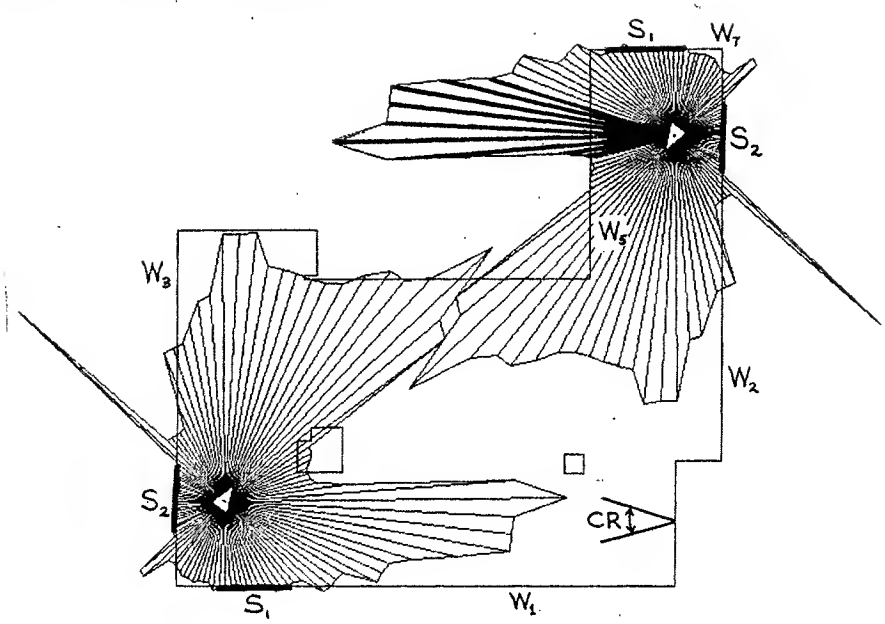


Figure 11: *These are two feasible localizations based on the interpretations* $[(S_1\ W_1)(S_2\ W_3)]$ *(lower left) and* $[(S_1\ W_7)(S_2\ W_2)]$ *(upper right). Given the cone of reflection as shown, the upper right localization would fail the sonar barrier test because the heavy black segments apparently penetrate wall* W_5 *from inside the cone of reflection.*

error"), since we are checking the error of sonar rays that we know to be almost normal to a wall (recall section 2.1.1).

The program would be highly inefficient if it carried out the search for "penetrating rays" in exactly the manner just described. Instead, the program first finds all walls containing at least one point such that a ray drawn from the robot location to the point lies within the CR for the wall. Then it checks only the sonar rays that fall inside the CRs of those walls. In Figure 12, for example, only the walls marked in heavy black would be chosen for the test, and only the heavy black sonar rays would be inspected for penetration of them.

The sonar barrier test can be very effective at pruning incorrect localizations that survive the local constraints and the global model test. It is common to have more than ten localizations that satisfy all of the local constraints and the global model test, with only one of them passing the sonar barrier test. This can happen, for example, when the robot is located near an isolated corner in the building. In this case the sonar segments often form a small L-shaped pattern. This arrangement of sonar segments does not help to select a particular corner in the building, since all corners are locally identical. The sonar barrier test uses the global information contained in the rest of the sonar contour to determine which corner the robot occupies.

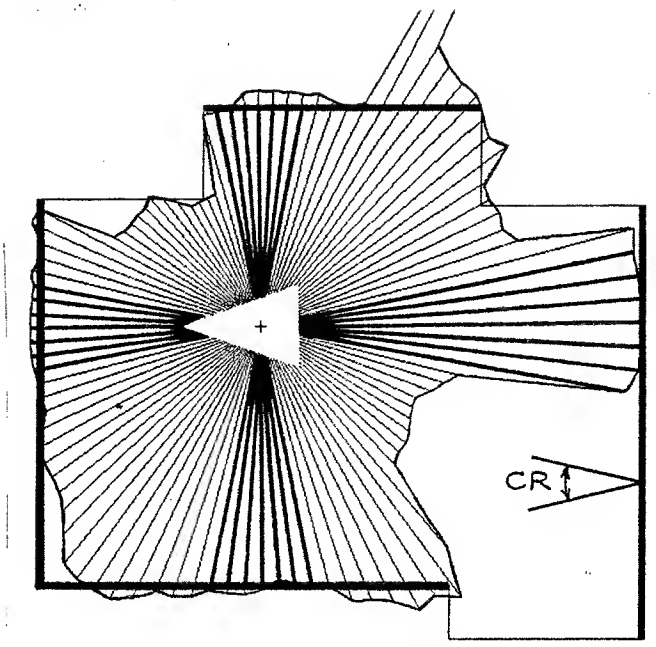


Figure 12: *Given the sonar contour and cone of reflection as shown, the sonar rays and walls drawn in heavy black would be used in the sonar barrier test.*

2.7 Ignoring Clutter and Other Bad Data

Sonar contours often contain a sonar segment that is not the sonar image of any wall included in the room model. Such a sonar segment could be produced by sonar noise or clutter, as described in section 2.2. It is impossible to find an interpretation that assigns such a sonar segment to a wall in a geometrically consistent way. Unfortunately, it is also impossible, using the process described so far, to distinguish between a violation of the local constraints due to an unfeasible wall assignment and a violation due to the presence of a clutter segment.

It may still be possible to find an interpretation of all the sonar segments, including the clutter segments, that is consistent with the local pruning constraints. In fact, it is even possible, by a fortuitous alignment of the data, for interpretations involving clutter segments to pass the global model test. However, such "freak" interpretations are almost always eliminated by the sonar barrier test.

We can assume, then, during this discussion, that any clutter segment will cause *all* interpretations to be inconsistent (except in rare cases). This poses a serious problem for our approach, as it is described so far, since all interpretations would be eliminated if just one of the sonar segments was clutter.

A straightforward way of handling this problem would be to apply the matching process to all subsets of the set of sonar segments, which would guarantee that a clutter-free set of sonar segments would be considered (if one existed). But, of course, this approach wastes much work determining the feasibility of the same partial interpretations. There is a way, however, to consider all subsets of the data without wasting the work of testing partial interpretations. This method was introduced in [Lozano-Pérez & Grimson 84], and is described as follows:

Consider the addition of one more branch to each node of the IT (see Figure 13).

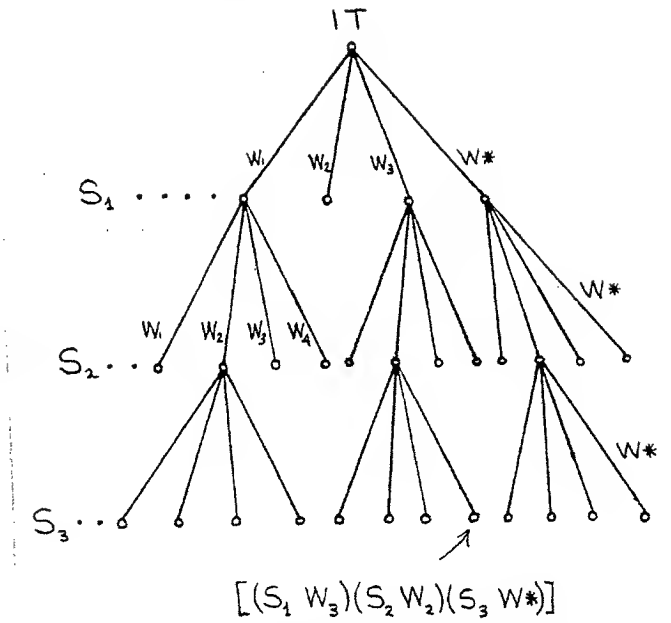


Figure 13: *The IT with the null wall branch added*

This branch represents the possibility that the sonar segment for that level of the IT is the image of a piece of clutter, i.e., that it should not be assigned to *any* wall. We will call this branch the *null wall*, or $W*$. The remainder of the process operates as before except that, when applying the local constraints, the null wall acts as a "wild card," i.e., it survives *all* of the local constraints, so that assigning a sonar segment to $W*$ never causes the failure of an interpretation. Thus, at every node that we visit in the IT, we assign the sonar segment under consideration to the null wall, to represent the possibility that the sonar segment is clutter.

It is easy to see that if an interpretation is admissible, the process described above will generate all subsets of this interpretation as leaves of the tree. This is true of partial interpretations as well as full interpretations. Every combination of assignments of the null wall to the sonar segments will still produce a valid interpretation, which guarantees that if any subset of the data points is valid, then a valid interpretation will be obtained as a leaf.

However, the null wall feature by itself greatly decreases the efficiency of the algorithm, since it causes the generation of all subsets of valid interpretations. We would rather generate only the interpretations that are consistent with as many as possible of the sonar segments. The following method guarantees that we find only the most complete interpretations, where "most complete" means "containing the fewest null-wall pairings."

The IT is explored in a depth-first fashion, with the null wall considered last when expanding a node. In addition, the global model test is applied to any interpretation that is a leaf of the IT. Now, suppose we use a global variable, called $BEST$, to record the number of non-null pairings that occur in the most complete interpretation found so far. As we expand the IT, we should assign a sonar segment S_i to the null wall *only if* $m + (s - i) \leq BEST$, where m is the number of non-null pairings in the current

node, s is the total number of sonar segments, and i is the depth of the current node in the IT. Otherwise, the number of non-null pairings in interpretations at all the leaves below this node will be less than that of the most complete interpretation already found. If we initialize $BEST$ to some non-zero value (usually two, since a unique localization requires at least two sonar segments), then only interpretations with this number of non-null pairings or greater will be found. Each time a more complete interpretation is found, the value of $BEST$ is incremented, thus ensuring that we find the most complete interpretation for the data. Note that if $BEST$ ever reaches s, then no null-wall assignments will have been made.

2.7.1 An Added Heuristic

In the rare event that all of the above procedures fail to produce a unique interpretation, then the interpretation that maximizes the sum of the lengths of sonar segments in non-null pairings is chosen as the final answer. This action expresses our preference for interpretations based on long sonar segments, which are less likely to be clutter segments. Note that the occurrence of multiple interpretations is still possible.

3 Results

The algorithm described in this paper has been run on real sonar data. The following section describes some of the results from these experiments.

3.1 The Experimental Setting

Sonar data was obtained from three different rooms in the Artificial Intelligence Laboratory at MIT. The room outlines are shown in Figures 14, 15, and 16. Next to each room outline is a photograph taken from inside the room. The photographs are intended to give the reader a rough idea of how cluttered and "real" these rooms are.

Figures 17 through 24 show some typical results from each of the three rooms. The robot's actual position was measured by hand with a tape measure and a protractor; it is accurate to within an inch or two and about five degrees. The surviving interpretations and their corresponding configurations are printed directly below each drawing. The actual configuration is also printed. In each drawing, the robot's actual configuration is represented by the outline of a triangle with a small circle inside it. The robot's sonar-determined configuration is represented by a solid white triangle with a cross in the center. Thus, the accuracy of a localization can be judged either by reading the printed numerical results or by observing the alignment of the triangles, the cross, and the circle.

The program parameters were the same for all of these results. They are not optimal, since they were chosen based upon only a small amount of experimentation

and the author's estimates of sonar range data errors. In terms of variables mentioned previously in the text, the program parameters were:

- $E = 1.3$ ft. (maximum error for a sonar ray endpoint),
- $E_\phi = 10$ degrees (maximum angular error for a sonar segment),
- $N = 7$ (minimum number of points in a sonar segment),
- $\delta = 0.5$ ft. (maximum perpendicular deviation for points in a sonar segment),
- $\gamma = 7$ degrees (CR half-angle),
- $E_n = 0.7$ ft. (maximum error for a sonar ray known to be nearly perpendicular to a wall)

Note that satisfactory results were obtained by assuming that all walls have a CR half angle γ equal to that of the most reflective surface that could possibly be encountered, namely, a smooth glass window (for which $\gamma < 10$ degrees).

The program has been run on 24 sonar contours so far. Seventeen of the sonar contours yielded a localization that was correct to within one foot in any direction and about five degrees in orientation. Two localizations were dead wrong, i.e., they were in error by more than one foot. The five sonar contours that yielded no localization failed for one of of the following reasons:

- the sonar contour contained only one sonar segment, or none at all,
- the sonar segments were either parallel or paired with parallel walls, a situation which can not produce a unique localization.

We have seen, both intuitively and from experience with the algorithm, that *it is unlikely that a localization based on a correct interpretation will be in error by a large amount.* This means that dead-wrong localizations are usually based upon incorrect interpretations. Since incorrect interpretations usually result only from "freak" alignments of the data, incorrect localizations also tend to be completely anomalous, having very large errors. This behavior could be desirable in the localization module of a real robot navigation system, since proposed configurations that are extremely different from recently determined configurations could be dismissed as obviously wrong. Large localization errors could be used as an indication that the robot should move slightly and try the algorithm again, or use another sensing approach.

The program was developed in Lisp on a Symbolics 3600 Lisp Machine. It usually takes about 5 seconds (after data acquisition) to localize the robot, but it has taken as long as 15 seconds for sonar contours containing a large amount of clutter.

4 Conclusions

Mobile robot localization can be performed quickly and reliably with a low-resolution, noisy rangefinder. The first step in the process is to reduce the size of the set of possible robot configurations by considering the possible pairings of wall-like data features with walls in the room model, pruning inconsistent pairings using local geometric constraints. The key to rejecting incorrect configurations that often persist in the resulting set of possible configurations is to exploit global geometric constraints, derived from simple physical laws, on the shape of a real sonar range contour.

The sonar data shown in this paper is so noisy and of such low resolution that it nearly constitutes a "worst case scenario" for range data. The author believes that the program's performance would be greatly enhanced by the use of a high-resolution laser rangefinder, such as the one described in [Jarvis 83].

5 Acknowledgments

The author wishes to thank Tomás Lozano-Pérez for his insightful guidance during the author's preliminary work on this subject as an undergraduate at MIT. Also, thanks to both Tomás Lozano-Pérez, Peter Gaston, and Eric Grimson for developing the basic ideas upon which much of this paper is based.

References

Ballard, Dana H., and Brown, Christopher M. *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

Drumheller, Michael. "Robot Localization Using Range Data: Achieving Accuracy Despite Noise and Clutter," *S.B. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering*, Cambridge, MA, May 1984.

Gaston, Peter C., and Lozano-Pérez, Tomás. "Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6, 3*, May 1984, p.257-266.

Giralt, Georges, and Sobek, Ralph, and Chatila, Raja. "A Multi-Level Planning and Navigation System for a Mobile Robot: A First Approach to Hilare," *Proceedings of the 6th International Joint Confererence on Artificial Intelligence*, Tokyo, August 1979.

Grimson, W. Eric L. "The Combinatorics of Local Constraints in Model-Based Recognition and Localization From Sparse Data," *MIT A.I. Lab, A.I. Memo 763* ,

Cambridge, MA, April 1984.

Grimson, W. Eric L., and Lozano-Pérez, Tomás. "Model-Based Recognition and Localization From Sparse Range or Tactile Data," *MIT A.I. Lab, A.I. Memo 738*, Cambridge, MA, August 1983.

Jarvis, R. A. "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5, 2*, March 1983, p.122-139.

Lewis, R. A., and Johnston, A. R. "A Scanning Laser Rangefinder for a Robotic Vehicle," *Proceedings of IJCAI-5*, 1977.

Lozano-Pérez, Tomás, and Grimson, W. Eric L. "Recognition and Localization of Overlapping Parts From Sparse Data," *Proceedings of the 2nd International Symposium on Robotics Research*, Kyoto, Japan, August 1984. (To be published by MIT Press, Cambridge, MA.)

Massa Products Corporation, anonymous author. "Model E-220 Ultrasonic Ranging Module," *technical brochure*, Hingham, MA 02043.

Miller, David. "Two Dimensional Mobile Robot Positioning Using Onboard Sonar," *Proceedings of the 9th William T. Pecora Memorial Remote Sensing Symposium, IEEE, USGS, NASA, ASP*, Sioux Falls, SD, October 1984, p.362-369.

Moravec, Hans P. *Robot Rover Visual Navigation*, UMI Research Press, Ann Arbor, Michigan, 1981.

Polaroid Corporation, anonymous author. "Ultrasonic Ranging System", *technical brochure. Ultrasonic Ranging Marketing division*, Cambridge, MA 02139.

Thompson, A. M. "The Navigation System of the JPL Robot," *Proceedings of IJCAI-6*, p.335-337, Tokyo, 1979.

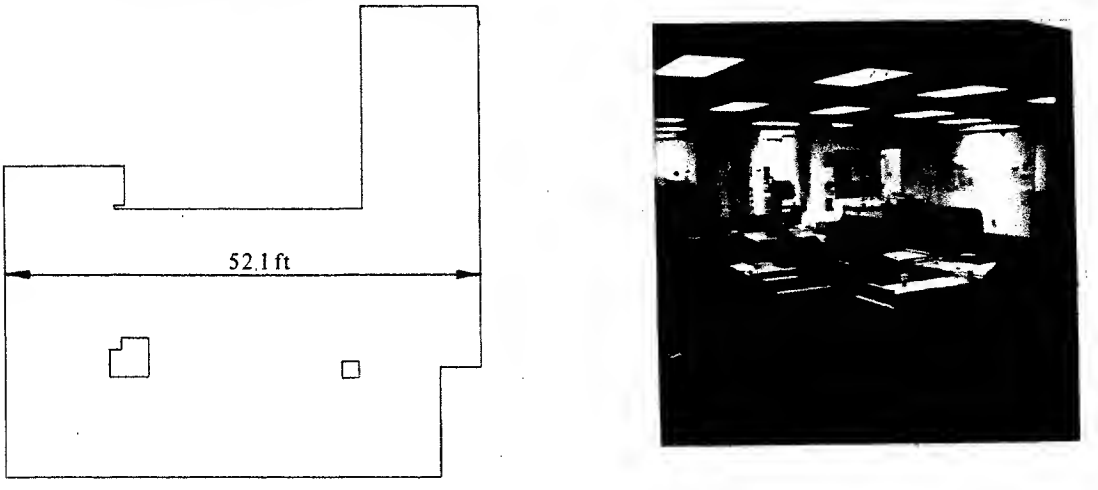


Figure 14: *Outline and photograph of the largest room tested, called BIG-RM.*

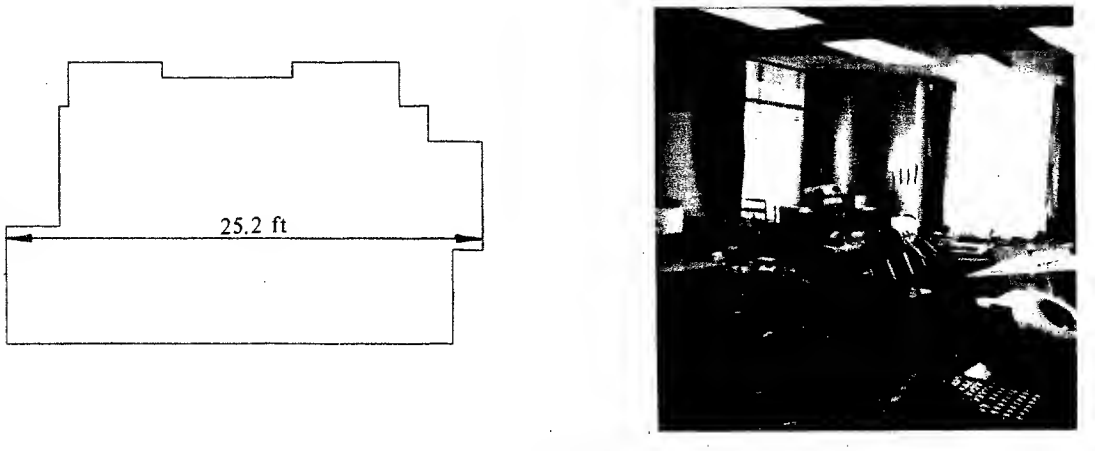


Figure 15: *Outline and photograph of the medium-sized room (MED-RM).*

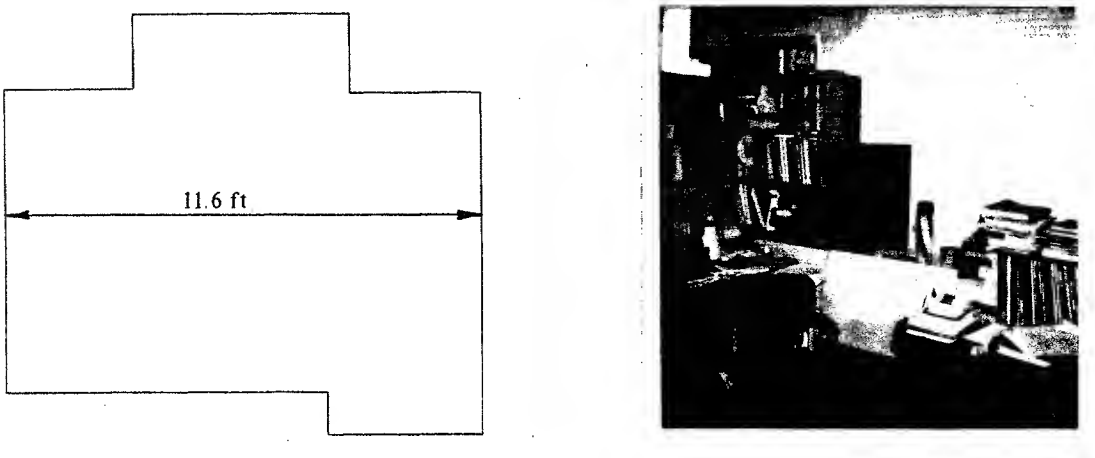


Figure 16: *Outline and photograph of the smallest room tested (SMALL-RM).*

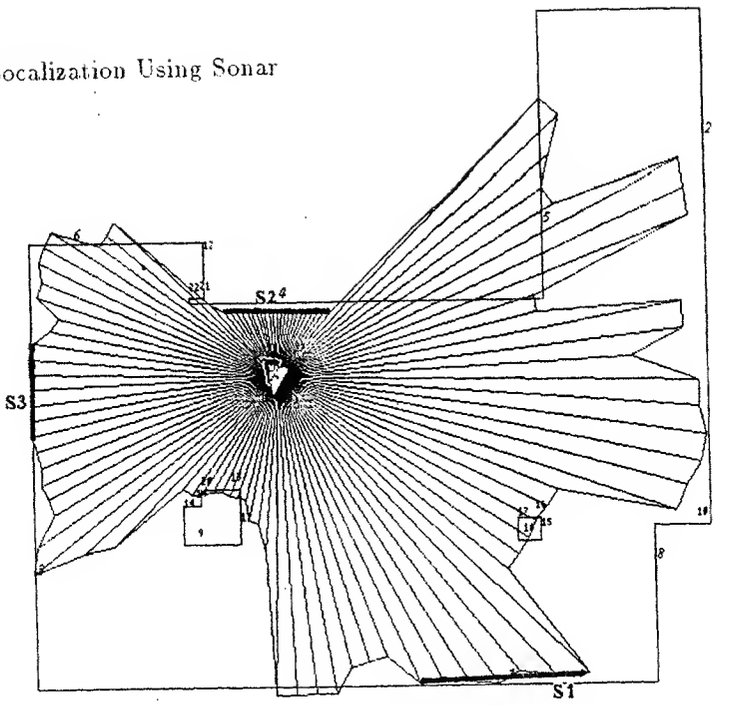


Figure 17: *Typical solution for BIG-RM. Note that the lower portion of the sonar contour contains large errors due to edge-of-beam reflection. (Recall Figure 4.)*

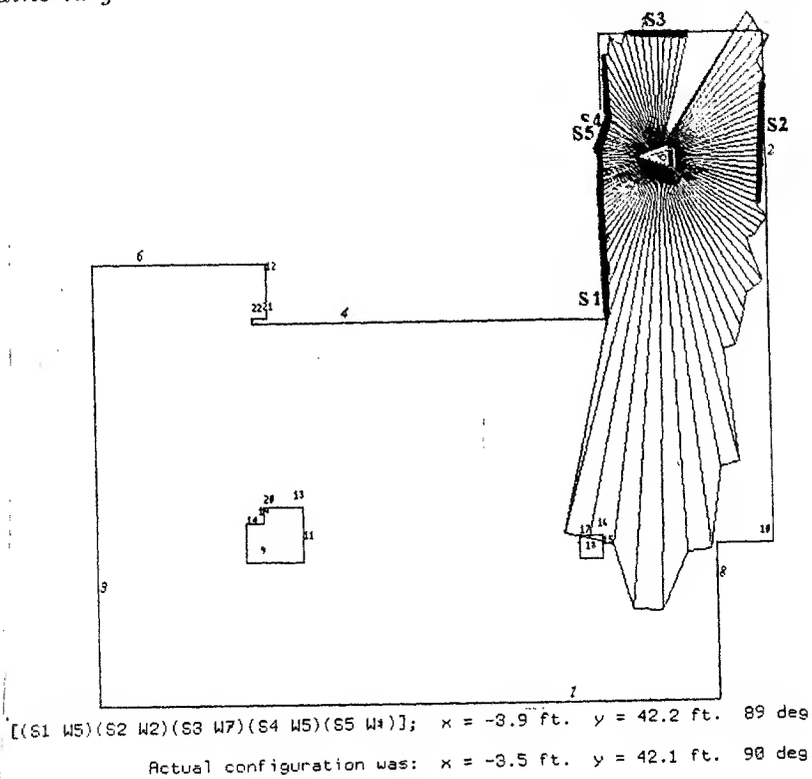
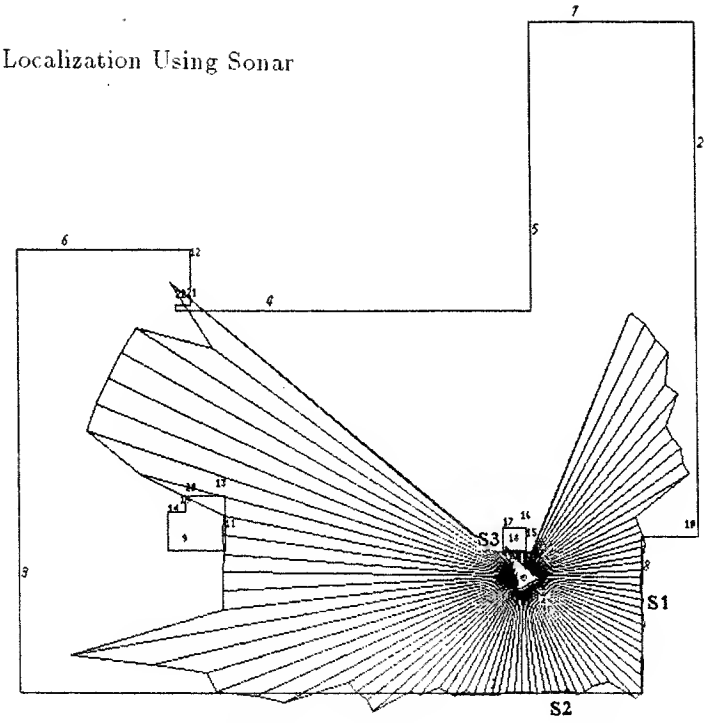


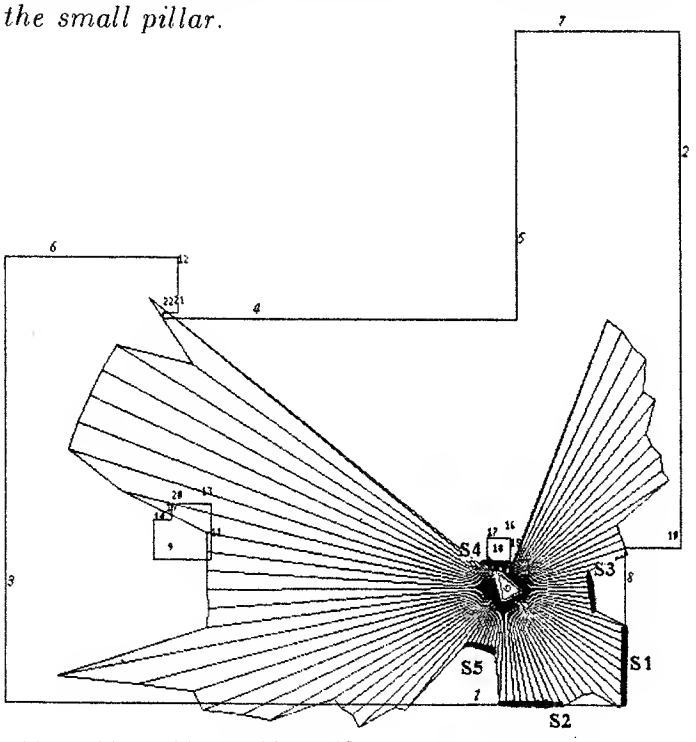
Figure 18: *In this example, the indentation in the sonar trace was produced when the beam just barely caught the top of the head of a small person standing nearby. The program successfully ignored the clutter. Note that the lack of global similarity between the sonar contour and the room outline does not affect the localization process.*



[(S1 W8)(S2 W1)(S3 W18)]; x = -9.1 ft. y = 9.0 ft. 32 deg.

Actual configuration was: x = -9.0 ft. y = 9.0 ft. 30 deg.

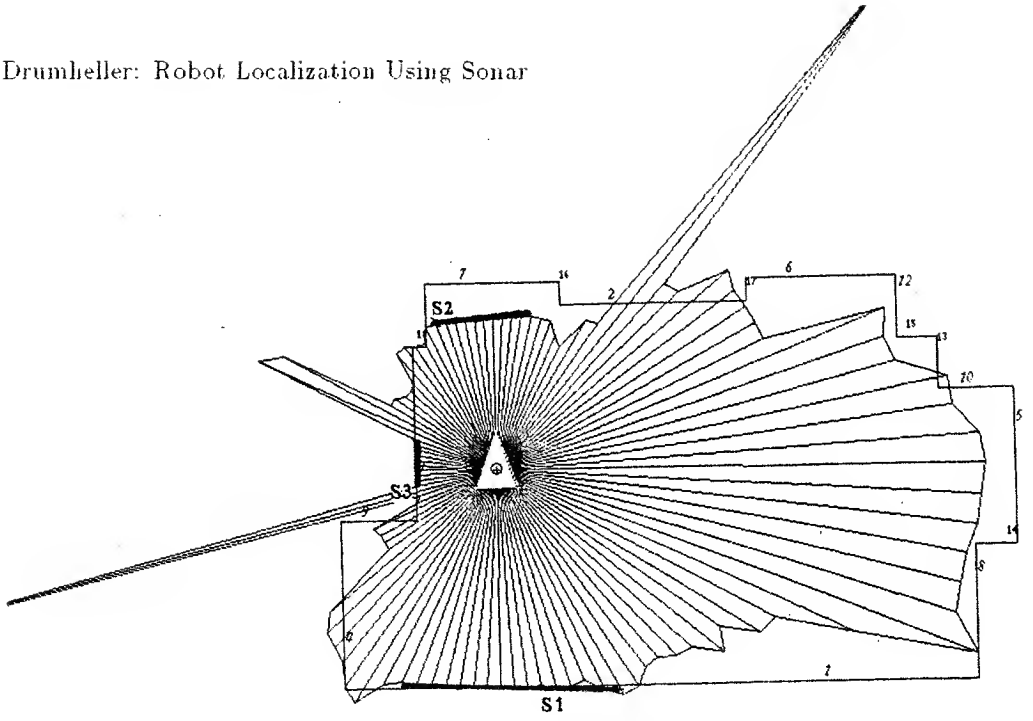
Figure 19: *In this localization, the segments S_1 and S_2 serve to select any corner configuration. The lower right configuration is selected because it causes S_3 to fit squarely against the small pillar.*



[(S1 W8)(S2 W1)(S3 W*)(S4 W18)(S5 W*)]; x = -9.2 ft. y = 9.0 ft. 33 deg.

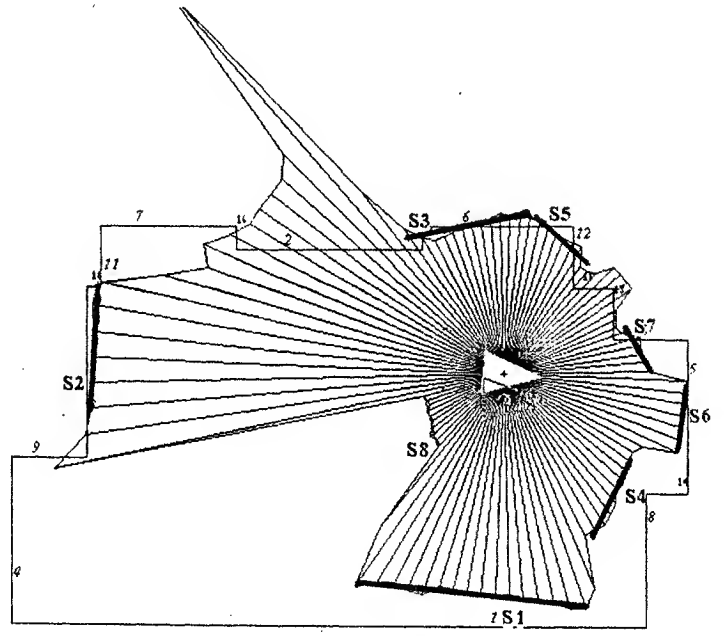
Actual configuration was: x = -9.0 ft. y = 9.0 ft. 30 deg.

Figure 20: *This sonar contour is similar to the contour in Figure 19, except for some deliberately introduced clutter (S_3 and S_5). The clutter segments were successfully ignored.*



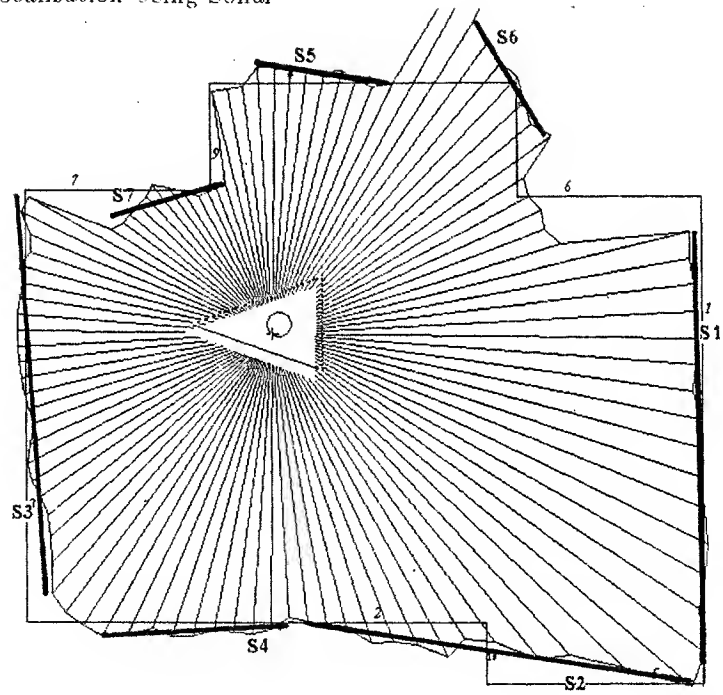

```
[(S1 W1)(S2 W*)(S3 W3)];  x = 5.8 ft.  y = 8.1 ft.  0 deg.
Actual configuration was:  x = 5.8 ft.  y = 8.1 ft.  0 deg.
```

Figure 21: *A typical result from inside MED-RM. Notice the prominent horns caused by false reflections.*



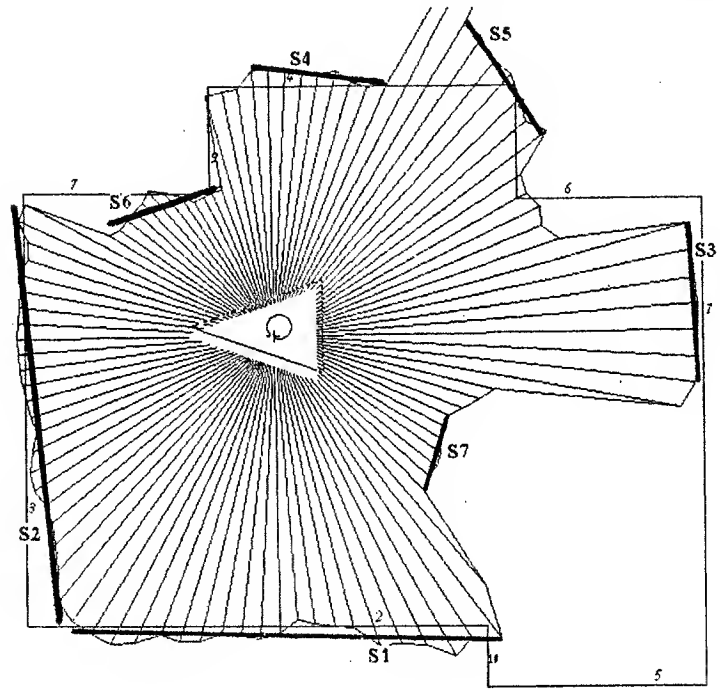

```
[(S1 W*)(S2 W3)(S3 W6)(S4 W*)(S5 W*)(S6 W5)(S7 W*)(S8 W*)];  x = 18.3 ft.  y = 9.4 ft.  -96 deg.
                                Actual configuration was:  x = 18.3 ft.  y = 8.5 ft.  -90 deg.
```

Figure 22: *In this case S_1 should have been assigned to the lower wall. However, the resulting interpretation would have been less complete, since almost every other segment would have been assigned to the null wall.*




```
[(S1 W1)(S2 W*)(S3 W3)(S4 W2)(S5 W4)(S6 W*)(S7 W*)];  x = -7.4 ft.  y = 6.0 ft.  89 deg.
                                Actual configuration was:  x = -7.2 ft.  y = 6.2 ft.  90 deg.
```

Figure 23: *A typical result from inside SMALL-RM.*




```
[(S1 W2)(S2 W3)(S3 W1)(S4 W4)(S5 W*)(S6 W*)(S7 W*)];  x = -7.4 ft.  y = 6.0 ft.  91 deg.
                                Actual configuration was:  x = -7.2 ft.  y = 6.2 ft.  90 deg.
```

Figure 24: *The same sonar contour as in Figure 23, with a tall person standing nearby in an attempt to inhibit localization. The clutter was successfully ignored.*